VLSI IMPLEMENTATION OF A RANDOM NUMBER GENERATOR USING A PLURALITY OF SIMPLE FLIP-FLOPS

This application claims priority to Provisional Serial No. **60/454,838** filed **March 14, 2003**.

FIELD OF THE INVENTION

[0001] The invention relates to the phenomenon of metastability and the effect of metastability on semiconductors. More particularly, the present invention relates to the use of metastability in the field of random number generators by creating a random number generator that makes use of the phenomenon via a plurality of flip-flops.

BACKGROUND ART

[0002] Latches and flip-flops are widely used in all types of electronic devices for counting, sampling, and storage of data. There are a number of different types of flip-flops named after their primary function, such as D-type flip-flops (data), J-K flip-flops (J and K inputs), and R-S flip-flops (having R and S latches, standard for "reset; and "set". D flip-flops are clocked flip-flops having one clock pulse delay for its output.

[0003] Conventional flip-flops, such as D-type, can be used to detect the logic state of an asynchronous digital signal with timing relative to the clock signal that is non-periodic.

[0004] However, the operating conditions of the flip-flops can be violated because hold times and setup times are not always consistent with the specifications (such as provided in the data sheets) of the flip-flops used. The violation of the operating conditions of the flip-flops can cause them to go into an unstable (metastable) state that can affect the entire operation of the linked systems. Metastability can occur when both inputs to a latch are set at a logic high (11) and are subsequently set at a logic low (00). Metastability can cause the latch outputs to oscillate unpredictably in a statistically known manner. Such metastable values are then detected by other circuitry as different logic states.

[0005] It has been found that intentionally inducing metastability provides the ability to harness the unpredictability of metastable flip-flops outputs as a random number generator.

[0006] For example, as shown in Fig. 1, a latch is realized with cross connected NAND gates 115,120. The flip-flop 110 drives this latch, It receives its clock input from clock oscillator 105 through the clock input 106 of flip-flop 110, and the inverting output –Q is connected to the D input, which shapes the clock signal to square-wave. The Q output 107 is connected to both of the NAND gates 115, 120 via delay devices, 112, 114, respectively. If the two NAND gates 115, 120 were truly identical, there would be no need for the delay devices to achieve the highest probability to get the flip-flop formed by the NAND gates 115, 120 to become metastable. However, the NAND gates will ordinarily differ somewhat, and their speed difference will influence the number of times metastability occurs in a time interval.

[0007] In VLS integrated circuits there have been attempts at tunable delay by using single tapped-buffer chains, but their implementation has not been practical. The delay resolution has been too course for the dynamic fine tuning required to achieve the highest frequencies at which metastability occurs. Delays were also designed by the introduction of long wires of various lengths, which increased design expense and has been found to be difficult to control using automatic layout tools and standard element libraries.

[0008] Current designs of physical (true) random number generators based on flip-flop metastability use single tapper-buffer-chains fixed delay values between their inputs to violate setup and hold timings, in order to provoke metastability. Eventually, the metastable state resolves to some logic level, which is effectively random, depending on the internal noise of the flip-flops. However, the fixed delay values used by the prior art can cause the random number generator to be susceptible to environmental changes. In addition, fixed delay values at large manufacturing variations can make the circuit not work at all or not work at optimal speed.

SUMMARY OF THE INVENTION

[0009] The present invention provides a random number generator, and a method of random number generation by exclusive-or-ing (XOR-ing) the output of a large number of individual flip-flops. The variation of the physical layout of the flip-flops ensures the necessary variance of the delays. The present invention eliminates the requirement of explicitly designing the circuit for different delay values, as shown, for example, by the introduction of delay units 112 and 114 in Fig. 1.

BRIEF DESCRIPTION OF THE DRAWING

[0010] Fig. 1 illustrates a prior art random number generator using a flip-flop and fixed delays.

[0011] Fig. 2 illustrates a prior art cascaded of D-type flip-flops to detect metastability.

[0012] Fig. 3A shows a first way that a variable delay can be introduced into the circuit arrangement by including a buffer at some of the data or clock inputs of the circuit.

[0013] Fig. 3B shows a second way that a variable delay can be introduced into the circuit arrangement by including a buffer at some of the cross-connected outputs.

[0014] Fig. 3C shows a third way that a variable delay can be introduced into the circuit arrangement by capacitive loading of some of the gates.

DETAILED EMBODIMENTS

[0015] It is understood by persons of ordinary skill in the art that the types of gates shown herein below were selected for explanatory purposes, and there can be different arrangements of different type of gates (NAND, NOR, XOR, etc.) in terms of size, function and connectivity that fall within the spirit of the invention and the scope of the appended claims.

[0016] Fig. 2 shows a conventional cascading of several flip-flops 210, 220, 230. We know that the probability of the circuit to have a stable output resulting in a 0 or 1 even at metastable input is quite high. If the input to the circuit comes from a metastable flip-flop, the output signal is random, most of the time at the standard logic levels.

[0017] However, if the input to the circuit does not come from the output of a metastable flip-flop, the output is deterministic in that it is directly derived from the clock 205 and the input applied at the gate D of flip-flop 110.

[0018] Rather than force a single flip-flop into a metastable state, It is proposed by the inventor to use a large number (e.g., several thousand) of slightly different flip-flops. One or a few of them become metastable, producing random output. By XOR-ing all of their outputs, which is equivalent to calculating the parity of all the outputs, the invention provides a random signal without knowing which of the particular flip-flops are behaving randomly and which are acting deterministically.

[0019] There are many different ways to implement the instant invention, and it should be understood that the following examples are shown for purposes of illustration and not for limitation.

[0020] With regard to Fig. 3A-3C, three techniques are shown to partition the flip-flops into groups (304,314,324; 334,344,354; 374,384,394). In each group the DATA and CLK inputs are connected to the same signal source, preferably a digital square wave generator.

[0021] As shown in Fig. 3A, the first group 304 comprises a plurality of flip-flops formed by NAND gates 305 and 310 without any buffers inserted between the data and clock signals. However, the second group 314 comprises of a plurality of flip-flops formed by NAND gates 315, 320, and have buffers 317 inserted between the data line and one of the flip-flops (in this case 320). The third group 324 comprises a plurality of flip-flops formed by NAND Gates 325,330 having their clock input connected through buffers 327. This way the layout dependent delay values for the flip-flops cause offsets varying around 0, + gate delay or – one gate delay.

[0022] The outputs of the three groups are connected to an exclusive OR gate network 331, that effectively calculates the parity of all the outputs. The output of the XOR gate network is input to a latch 332. For example, the first 2 flip-flop outputs get XOR-ed, their output and the third flip-flop out put XOR-ed with a second XOR gate, and so on, until the last flip-flop output is XOR-ed with the output of the previous XOR gate. Since XOR is addition modulo 2, XOR-ing many digital signals is the same as calculating their sum modulo 2, which is the parity of their sum.

[0023] As shown in Fig. 3B, the flip-flops here are realized by two groups of NAND gates. The first group comprises a plurality of flip-flops realized by NAND gates 335 and

340 have a delay buffer 342 in one of the cross connection lines, while another group of flip-flops are realized by NAND gates 345, 350, having delay buffers 347 in the other cross-connection line, while the third group of flip-flops comprises a plurality of NAND gates 355, 360 have no delay buffers.

[0024] Again, the outputs of the three groups are connected to an exclusive OR gate network 331, effectively calculating the parity of all the outputs.

[0025] Finally, Fig. 3C shows yet another way to arrange the flip-flops, in this case having a capacitive load 365, 370 added to one of the data or clock inputs of one of the NAND gates 375,390 in the form of a multi-input gate. A third set of gates 397,399 would have no capacitive load at all attached thereto. The capacitance of the parallel connected inputs of multi-input gates slows down the signal changes, effectively introducing some delays. The output of these "load" gates 365, 370 need not be connected anywhere and may float. In the instance of the flip-flops formed by instances of NAND gate 390, the cross connected output of gate 395 is connected to the inputs of multi-input gate 370. The coupling of the multi-input gate affects the circuit timing.

[0026] There are a number of variations possible according to the circuit point the capacitive load is attached to: the input of the upper (375,390) or lower (380,395) NAND gates.

[0027] Similar to the other arrangements, the outputs of the pairs of gates are connected to an exclusive OR gate network 331 that effectively calculates the parity of all the outputs.

[0028] It should be understood that there could be literally thousands of flip-flops being partitioned into groups as shown in Fig. 3A, 3B or 3C. It is also possible to use a different number of flip-flops for each group, and implement only some of the possible groups in one instance of the invention. It is even possible to intermingle combinations of all three types of arrangements shown in respective Figures 3A through 3C. Finally, while NAND gates are shown, a person of ordinary skill in the art knows that equivalent Boolean configurations could be used. In fact, an equivalent arrangement of one of the pairs of NAND gates shown above, that is cross connected with another NAND gate, there could also be another way to introduce delay without the need for explicitly designing the circuit to have different delay values.